KATHOLIEKE UNIVERSITEIT
LEUVEN

Faculty of Business and Economics

Differential evolution to solve the lot size problem in Stochastic production systems

Kris Lieckens and Nico Vandaele

DEPARTMENT OF DECISION SCIENCES AND INFORMATION MANAGEMENT (KBI)

KBI 1121

Differential Evolution to Solve the Lot Size Problem in Stochastic Production Systems

K. Lieckens[ab1] and N. Vandaele[ab2]

[a]Faculty of Business and Economics, Research Center for Operations Management, K.U.Leuven, Belgium

[b]Faculty of Applied Economics, K.U.Leuven Campus Kortrijk, Belgium

August 16, 2011

Abstract

An Advanced Resource Planning model is presented to support optimal lot size decisions for performance improvement of a production system in terms of either delivery time or setup related costs. Based on a queueing network, a model is developed for a mix of multiple products following their own specific sequence of operations on one or more resources, while taking into account various sources of uncertainty, both in demand as well as in production characteristics. In addition, the model includes the impact of parallel servers and different time schedules in a multi-period planning setting. The corrupting influence of variabilities from rework and breakdown is explicitly modeled. As a major result, the differential evolution algorithm is able to find the optimal lead time as a function of the lot size. In this way, we add a conclusion on the debate on the convexity between lot size and lead time in a complex production environment. We show that differential evolution outperforms a steepest descent method in the search for the global optimal lot size. For problems of realistic size, we propose appropriate control parameters for the differential evolution in order to make its search process more efficient.

Keywords: Production Planning, Lot Sizing, Queueing Networks, Differential Evolution

1 Introduction

Even in today's state-of-the-art production systems that have implemented lean manufacturing techniques like 6-sigma and SMED analysis, the amount of time to switch the process before the production of parts from another product family takes place, may still remain significant. Given this lean approach, the performance of the production system with respect to lead times and setup related costs can be improved by optimizing the lot size quantities of each product (Karmarkar et al., 1985). Since the impact of these lot size decisions on the lead time is a difficult operational problem, it has received much attention in production planning research (see below). It involves a trade-off between capacity and lead time.

Getting started at large lot sizes, decreasing the lot size has two opposing effects. First of all, more resource capacity is consumed due to an increased number of setups. For this increased

[1]Corresponding author. Phone: +32 16 32 67 58. Email: kris.lieckens@econ.kuleuven.be; kris.lieckens@kuleuven-kortrijk.be

[2]Phone: +32 16 32 67 51. Email: nico.vandaele@econ.kuleuven.be; nico.vandaele@kuleuven-kortrijk.be

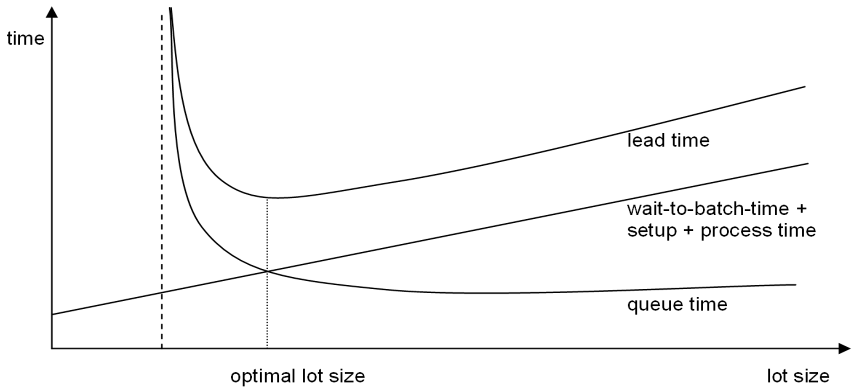

Figure 1: Convex Relationship between Lot Size and Lead Time.

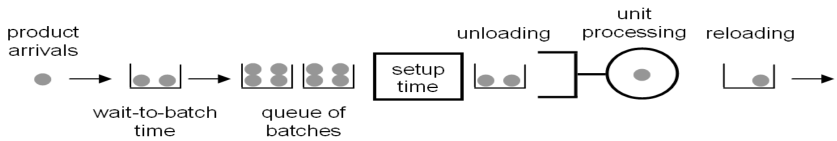

Figure 2: Sequential Production System.

utilization level, lead times increase in a nonlinear way: it is rather stable at low utilization levels, but it explodes with high utilizations. So the impact of a lot size reduction on lead time depends on the utilization level. On the other hand, decreasing the lot size has a positive effect on the lead time due to less wait-in-batch time (all batches are finished more rapidly). In addition, it takes less time at the first operation to form the batch (wait-to-batch-time). Opposite relationships are observed when the lot size is increased. It is the lot size quantity in combination with the other production characteristics that determine which one of these opposing effects is dominant. Both effects are responsible for the convex relationship between lot size and lead time (see Figure 1). It typically applies to operations such as molding, painting, bottling, etc., where parts are sequentially processed as in Figure 2. A delay due to setup is incurred before switching to a different product type and total process time grows proportionally with the lot size. This is in contrast to a simultaneous operation that works on several parts at a time like heating, and where a maximum number of parts can fit (see Figure 3). Although the process time is independent from the lot size, a similar convex relationship holds due to analogue queueing and wait-to-batching effects. The analysis is only slightly different and therefore we focus on the sequential case. For more details, we refer to Hopp and Spearman (2000).

The lot size decision is clearly a difficult problem, especially when multiple products, multiple machines, multiple operations and multiple time schedules characterize the production system in addition to various sources of process disruptions (e.g. stochastic demand, variabilities in production times, rework, breakdowns etc.). It is for this type of environment that we want to develop a production planning tool that supports decisions at an intermediate time horizon concerning the optimal lot size values, further referred to as the Advanced Resource Planning (ARP). Optimizing lot sizes is in this respect a key issue.

This paper extends on the work of Vandaele (1996) and Lambrecht et al. (1998). Since

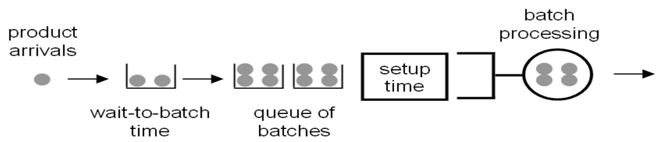

Figure 3: Simultaneous Production System.

their model is limited to single machine centers, we want to integrate parallel servers. Another extension relates to the integration of a quantitative measure for the impact of other operational issues like different time schedules, rework and machine breakdowns. In addition to a weighted average lead time as an overall performance measure for the production system, we develop here an alternative objective where the lot size impact is translated into a cost function that consists of setup and inventory costs.

However, the main focus of this paper is on the optimization of the ARP problem as described in Section 2, and on the convex relationship between the lot size and the objective function (either total lead time or cost). For the single product case and for an M/M/1 queue, Karmarkar (1987) has described an analytical proof. For the multi-product, multi-machine case, the complexity of an analytical analysis of the lot size problem (mainly caused by interdependencies between total lead times and the mix of product types) increases in such a way that in this general case, there are no exact closed-form analytical expressions for the optimal lot size; only approximations exist. In a multi-product, single stage setting with an M/G/1 queue, the expected queueing delay is quasi-convex in the lot size (Karmarkar et al., 1992). When lot size independent weights are used to obtain a weighted average total lead time of a batch, this lead time measure is also quasi-convex in the lot sizes (Kuik and Tielemans, 2004). As the production system becomes more complicated (multiple stages, multiple resources, multiple servers, ...), no proof on the convexity can be found in the literature. Authors who consider this type of systems like Lambrecht et al. (1998) were limited to postulate the convex behavior. This paper exploits a specific search characteristic of the Differential Evolution (DE) procedure to support evidence for this relationship. According to Lampinen and Zelinka (1999), Storn and Price (1997) and Babu and Angira (2002), DE is a powerful algorithmn in the search for global optimum in case of hard, nonlinear problems with integer variables, even when many local optima exist. For our problem setting, the performance of DE and results for different control parameter settings are compared with the best result obtained by the steepest descent method (SD) that is described in Vandaele (1996) and used in Lambrecht et al. (1998). We also show that the continuous treatment of discrete variables may lead to suboptimal results.

The research questions, for which findings are outlined in Section 5, are the following:

1. Is the differential evolution algorithm able to handle large real-life problems?
2. Does the differential evolution algorithm outperforms dedicated search algorithms like the steepest descent, in terms of objective value and computation time? In other words, is DE effective and efficient compared to SD?
3. What are efficient control parameters for the differential evolution algorithm?
4. Is there evidence for the generally postulated convex relationship between lot size and lead time in a multi-product, multi-machine production system?

5. Does the optimal set of lot size values differ for both objective functions (lead time and costs)?

The paper is organized as follows. We first describe the problem in more detail in Section 2. In Section 3 all the equations required for the lead time and cost objective in the ARP model are established. The optimization routines are presented in Section 4, followed by their results for real-life applications in Section 5. Conclusions are outlined in Section 6.

2 Model Description

ARP is developed for a production system where a mix of multi-products needs to be produced on multi-resources according to a first-come, first-served priority rule. Resources can be either machines or labor, both with single or parallel servers. For convenience, we only use the term 'machine' in this paper. The demand is based on forecasts and for firm customer orders, while the dynamic nature of the customer demand pattern is explicitly taken into account. Demand is independent and identically distributed, but we do not pose any restriction on the type of this distribution. Each product has its own specific bill of processes, with a fixed sequence of operations on resources that is known prior to the shop release of the batch (i.e. deterministic routing). Here, setup and production processes are typically stochastic: their required times are also generally distributed. Another process characteristic is that a product's lot size quantity remains fixed along its route. As a result, transfer batching is not allowed, which means that the entire batch must be completed before items are released to the next operation. We further assume that setup times and costs are independent of the production sequence. This model can be executed in a repetitive and independent way for multiple planning periods on the intermediate time range, which makes it useful for the sales and operations planning process. These time buckets that constitute the planning horizon are long enough to ensure steady state conditions.

In order to determine the optimal lot sizes that minimize lead times or costs, we transform the production system into a queueing network where the parameters and the expected lead time are a function of the lot size (see Section 3). As a result, the utilization impact and the stochastic nature of the problem are included in the model, while an estimation for the lead times and inventory costs can be calculated. The queueing approach is similar to Lambrecht et al. (1998), who on their turn have included some additional features into queueing approximations found in the literature (Shanthikumar and Buzacott (1981), Buzacott and Shanthikumar (1985), Shanthikumar and Sumita (1988), Bitran and Tirupati (1988)). One of their main contribution is the approximation of the lead time distribution by a lognormal distribution, which is used to estimate a safety lead time such that customer orders are satisfied with a predefined service probability.

Once the optimal lot sizes are determined, the corresponding expected lead time can be obtained for each product. This is one of the main ARP outputs because it enables release date settings that will meet a specific due date under a given service level constraint by adding a safety lead time on top of the expected lead time. It creates a time window within which all the operations must be scheduled. Lambrecht et al. (1998) suggest a hierarchical approach to link separate applications into an integrated planning and scheduling system. In this paper, we focus on the lot size optimization process (Section 4) and a refinement of the underlying queueing model (Section 3).

3 Model Formulation

In this section, all the relationships are derived that are required to formulate the objective function for lead time estimation. Since this function will depend on the lot size quantities of the products, an optimization routine can be developed. The following indexes are used: multiple products ($p = 1, ..., P$), multiple machines ($m = 1, ..., M$) and for each product p, different operations ($o = 1, ..., O_p$). We define a binary parameter ς_{pom} that is equal to 1 if operation o for product p requires machine m, and 0 otherwise. When the notation for a product parameter is complemented by a tilde ($\tilde{}$), it represents the same parameter, but related to the entire lot size. Given different operational schemes (shift regime, working time, breakdowns, ...), there will be a need for a common time unit (e.g. hours) for all the time based parameters in the queueing model.

3.1 Availability

Since ARP will be used in real-life applications, the issue of different time schedules in a given period must be handled. Clearly, the occurrence of customer demand does not coincide with the manufacturing operating hours, and takes neither overtime, days-off and absenteeism of personel nor breakdowns and preventive maintenance of machines into account. Therefore, we express all queueing parameters on a common, continuous time scale (24 hours per day, 7 days per week). The advantage of this approach is that the outcome of the queueing model is easy to interpret: the delay is expressed in a number of calendar days, regardless of the available time for the respective resources in the underlying production system. To this end, we calculate the availability measure A_m, i.e. the percentage of time that each resource m is available for processing. When the breakdown pattern of a machine m is described by a Mean-Time-To-Repair $MTTR_m$ and a Mean-Time-Between-Failures $MTBF_m$, and the schedule of each machine m is given for the planning period (total number of working time units WT_m, i.e. in normal and overtime regime, and total number of time units for preventive maintenance PM_m), we then propose here below the equation for the machine availability during that planning period of continuous length with CT time units, also referred to as the time bucket

$$A_m = \frac{(WT_m - PM_m)(MTBF_m/(MTTR_m + MTBF_m))E_m}{CT} \tag{1}$$

where E_m is a percentage for the efficiency to handle various sources of losses in productive time not covered by the other measures. The planning horizon consists of multiple time buckets.

3.2 Arrival Process

Multiple products p are required at an average rate λ_p. This is the product specific demand quantity that we expect to arrive per time unit defined on a continuous time scale, and is equivalent to an expected time between demand occurrences of $IA_p = 1/\lambda_p$. Since this interarrival time is stochastic for each product p, we use a squared coefficient of variation (SCV) $c^2_{IA_p}$ to describe its degree of randomness. It is the machine m of the first operation ($o = 1$) that is affected by this demand variability. Subsequent operations will face a variability at their inbound that is determined by variability and utilization levels at all upstream stations (see Equation (9)).

Before these demand arrivals of product p are released into the shop floor, they are grouped by a quantity Q_p, the manufacturing lot size, which we assume to remain constant for all operations

o of product p. This batching process has several effects. First of all, the production system is also characterized by an average batch arrival rate $\tilde{\lambda}_p$ and a SCV of batch interarrival times $\tilde{c}^2_{IA_p}$ for each product p, which can be calculated by λ_p/Q_p and $c^2_{IA_p}/Q_p$ respectively. Another issue is the so-called wait-to-batch-time $WTBT_p$, a collection time to form a batch of size Q_p. This delay is only obeserved before the first operation as a batch is assumed not to be split when proceeding along the route, and is equal to

$$WTBT_p = \frac{(Q_p - 1)}{2\lambda_p} \tag{2}$$

because the first unit in a batch waits for $Q_p - 1$ other units to arrive and therefore waits $(Q_p - 1)/\lambda_p$ time units, whereas the last one does not have to wait at all to join the batch. A similar approach can be found in Vandaele et al. (2003). For our queueing network approach, these multi-product batch arrival processes have to be aggregated into a single batch arrival process at each machine m. This aggregate process is also characterized by an average aggregate batch arrival rate $\tilde{\lambda}_m$ and a SCV of the aggregate batch interarrival times $\tilde{c}^2_{IA_m}$. When we define the aggregate batch arrival rate of product p at machine m as $\tilde{\lambda}_{pm} = \sum\limits_o \tilde{\lambda}_p \varsigma_{pom}$, we have $\tilde{\lambda}_m = \sum\limits_p \tilde{\lambda}_{pm}$. This includes batch arrivals at machine m that are both internal, i.e. coming from another machine, and external, i.e. coming from the customer. The external aggregate batch arrival rate at machine m is defined as $\tilde{\lambda}'_m = \sum\limits_p \tilde{\lambda}_p \varsigma_{p1m}$. The value of $\tilde{c}^2_{IA_m}$ is derived in Section 3.3 because it depends on the variability impact of the production processes that are involved in a product's route. At this moment, we can only obtain an approximation for the SCV of the external aggregate batch interarrival times $\tilde{c}'^2_{IA_m}$

$$\begin{array}{ll} \tilde{c}'^2_{IA_m} \approx \frac{1}{3} + \frac{2}{3}\sum\limits_p \frac{\tilde{\lambda}_p \varsigma_{p1m}}{\tilde{\lambda}'_m} \tilde{c}^2_{IA_p} & \text{if } \sum\limits_p \varsigma_{p1m} \geq 2 \\ \tilde{c}'^2_{IA_m} = \tilde{c}^2_{IA_p} & \text{if } \sum\limits_p \varsigma_{p1m} = 1 \end{array} \tag{3}$$

where the weights 1/3 and 2/3 are discussed in Lambrecht et al. (1998). It is a specific case of a general approximation found by Albin (1981).

3.3 Production Process

Before being shipped to the customer, each product p typically requires one or more operations o that are performed on one or more machine types m. We allow for machine recurrences along a product's route. The machine m to be used for a particular operation o of product p is handled by the previously introduced parameter ς_{pom}. Next, we list for each product p and operation o the following production characteristics: expected setup time SU_{po} with SCV $c^2_{SU_{po}}$ and variance $\sigma^2_{SU_{po}}$, expected unit processing time PR_{po} with SCV $c^2_{PR_{po}}$ and variance $\sigma^2_{PR_{po}}$, expected unit process rate $\mu_{po} = 1/PR_{po}$, and a rework percentage of $rwrk_{po}$. Some input parameters related to a machine m have been listed in Section 3.1. One of the contributions of this paper is the extension of the queueing model as described in Lambrecht et al. (1998) towards parallel machines, referred to by the parameter s_m. All the required input data of the queueing model, for the arrival as well as the production process, are given at this point.

The first step to build the ARP model is the transformation of setup and process time characteristics (averages and variances) into effective measures using the availability A_m and the rework percentage $rwrk_{po}$ (Hopp and Spearman, 2000). We obtain effective averages of setup and process times as follows

$$\begin{aligned} SUe_{po} &= SU_{po} / \left(\sum_m A_m \varsigma_{pom} \right) \\ PRe_{po} &= PR_{po} / \left((1 - rwrk_{po}) \sum_m A_m \varsigma_{pom} \right) \end{aligned} \tag{4}$$

and their effective variances as

$$\begin{aligned} \sigma^2_{SUe_{po}} &= c^2_{SU_{po}} SUe^2_{po} \\ \sigma^2_{PRe_{po}} &= \frac{c^2_{PR_{po}} PRe^2_{po} + 2PRe_{po} \sum_m MTTR_m (1-A_m) \varsigma_{pom}}{1 - rwrk_{po}} + \frac{rwrk_{po} PRe^2_{po}}{(1-rwrk_{po})^2} \end{aligned} \tag{5}$$

Similar to the arrival process, we need an average and a SCV of the aggregate batch processing time on machine m. The expression for the average is a weighted average over all the products and all their operations on that machine m

$$\tilde{PR}_m = 1/\tilde{\mu}_m = \sum_p \frac{\tilde{\lambda}_{pm}}{\tilde{\lambda}_m} \sum_o \frac{\tilde{\lambda}_p \varsigma_{pom}}{\tilde{\lambda}_{pm}} \left(SUe_{po} + Q_p PRe_{po} \right) \tag{6}$$

where $\tilde{\lambda}_{pm}/\tilde{\lambda}_m$ is the probability that a randomly selected batch at the inbound of machine m is of product p, while its SCV becomes

$$\begin{aligned} \tilde{c}^2_m = & \left[\sum_p \frac{\tilde{\lambda}_{pm}}{\tilde{\lambda}_m} \sum_o \frac{\tilde{\lambda}_p \varsigma_{pom}}{\tilde{\lambda}_{pm}} \left(SUe_{po} + Q_p PRe_{po} \right)^2 \right] \tilde{\mu}^2_m - 1 + \\ & \sum_p \frac{\tilde{\lambda}_{pm}}{\tilde{\lambda}_m} \sum_o \frac{\tilde{\lambda}_p \varsigma_{pom}}{\tilde{\lambda}_{pm}} \frac{\left(\sigma^2_{SUe_{po}} + Q_p \sigma^2_{PRe_{po}} \right)}{\left(SUe_{po} + Q_p PRe_{po} \right)^2} \end{aligned} \tag{7}$$

Next, we determine the utilization ρ_m of machine m

$$\rho_m = \frac{\tilde{\lambda}_m}{\tilde{\mu}_m s_m} = \sum_p \sum_o \tilde{\lambda}_p \varsigma_{pom} \left(SUe_{po} + Q_p PRe_{po} \right) / s_m \leq 1 \tag{8}$$

where total setup time is added to total process time. This is called the adapted traffic intensity for machine m, and must be lower than 100%. The final input parameter to be derived for the queueing network model is the SCV of the aggregate batch interarrival times at machine m or $\tilde{c}^2_{IA_m}$, for which the variability of interdeparture times of batches leaving the upstream machine m' is to be known. Although Buzacott and Shanthikumar (1993) have presented several approximations for systems with multiple servers, it is reasonable to estimate it by the following linking equation (Hopp and Spearman, 2000)

$$\tilde{c}^2_{ID_{m'}} \approx 1 + \left(1 - \rho^2_{m'}\right) \left(\tilde{c}^2_{IA_{m'}} - 1 \right) + \frac{\rho^2_{m'}}{\sqrt{s_{m'}}} \left(\tilde{c}^2_{m'} - 1 \right) \tag{9}$$

We see that it is determined, not only by the utilization, but also by the production and arrival processes at the upstream machines. From Equations (3), (7) and (8), we know that the lot size decisions will have an impact on $\tilde{c}^2_{ID_{m'}}$. Furthermore, the arrival process at a downstream machine m depends on the fraction leaving machine m' and going towards machine m, $f_{m'm}$.

The SCV of the aggregate batch interarrival time at machine m coming from machine m', i.e. $\tilde{c}^2_{IA_{m'm}}$, is related to $\tilde{c}^2_{ID_{m'}}$ according to the equation (Shanthikumar and Buzacott, 1981)

$$\tilde{c}^2_{IA_{m'm}} = f_{m'm}\tilde{c}^2_{ID_{m'}} + (1 - f_{m'm}) \tag{10}$$

The expression for $\tilde{c}^2_{IA_m}$ is the sum of a weighted average of $\tilde{c}'^2_{IA_m}$ and $\tilde{c}^2_{IA_{m'm}}$

$$\tilde{c}^2_{IA_m} = \sum_{m'} \left(\frac{\tilde{\lambda}_{m'}}{\tilde{\lambda}_m} f_{m'm} \right) \tilde{c}^2_{IA_{m'm}} + \frac{\tilde{\lambda}'_m}{\tilde{\lambda}_m} \tilde{c}'^2_{IA_m} \tag{11}$$

which by using Equation (10) becomes

$$= \sum_{m'} \left(\frac{\tilde{\lambda}_{m'}}{\tilde{\lambda}_m} f_{m'm} \right) \left(f_{m'm}\tilde{c}^2_{ID_{m'}} + (1 - f_{m'm}) \right) + \frac{\tilde{\lambda}'_m}{\tilde{\lambda}_m} \tilde{c}'^2_{IA_m} \tag{12}$$

and when combined with Equation (9) results in

$$\begin{aligned} \approx\ & \sum_{m'} \left(\tfrac{\tilde{\lambda}_{m'}}{\tilde{\lambda}_m} f_{m'm} \right) \left(f_{m'm} \left[1 + \left(1 - \rho^2_{m'}\right) \left(\tilde{c}^2_{IA_{m'}} - 1 \right) + \tfrac{\rho^2_{m'}}{\sqrt{s_{m'}}} \left(\tilde{c}^2_{m'} - 1 \right) \right] + \right. \\ & \left. (1 - f_{m'm}) \right) + \tfrac{\tilde{\lambda}'_m}{\tilde{\lambda}_m} \tilde{c}'^2_{IA_m} \end{aligned} \tag{13}$$

Reorganization of Equation (13) when expressed as an equality, leads to M linear equations with M unknown parameters $\tilde{c}^2_{IA_m}$

$$\begin{aligned} & -\sum_{m'} \tilde{\lambda}_{m'} f^2_{m'm} \left(1 - \rho^2_{m'}\right) \tilde{c}^2_{IA_{m'}} + \tilde{\lambda}_m \tilde{c}^2_{IA_m} = \\ & \sum_{m'} \tilde{\lambda}_{m'} f_{m'm} \left(f_{m'm} \rho^2_{m'} \left(1 + \tfrac{\tilde{c}^2_{m'} - 1}{\sqrt{s_{m'}}} \right) + 1 - f_{m'm} \right) + \tilde{\lambda}'_m \tilde{c}'^2_{IA_m} \end{aligned} \tag{14}$$

To solve these equations and find values for $\tilde{c}^2_{IA_m}$, we have to obtain the transition matrix F, which is easy because of the assumed deterministic routings. The transitions consist of

1. the proportion of batches from outside the system (stage 0) and directed to the first machine in the routing (stage m)

$$f_{0m} = \tilde{\lambda}'_m / \sum_m \tilde{\lambda}'_m \tag{15}$$

2. the proportion of batches from machine m' (stage m') and directed to the next machine in the routing (stage m)

$$f_{m'm} = \sum_p \sum_o^{O_p - 1} \tilde{\lambda}_p \varsigma_{pom'} \varsigma_{po+1m} / \tilde{\lambda}_{m'} \tag{16}$$

3. the proportion of batches from the last machine in the routing (stage m) and directed to the outside of the system (stage 0)

$$f_{m'0} = \sum_p \tilde{\lambda}_p \varsigma_{pO_pm'} / \tilde{\lambda}_{m'} \tag{17}$$

At this point we are able to estimate the expected lead time for an operation o of product p, which takes place at machine m. Instead of using the Kraemer-Lagenbach-Belz approximation as in Lambrecht et al. (1998), we opt for the approximation from Whitt (1993). The reason is twofold: it is a GI/G/m-model that applies to a more realistic production system with parallel servers and it estimates the waiting time under heavy traffic conditions more accurately due to a correction factor ϕ. We refer to A for more details about this factor. By using Equations (6), (7), (8) and (14), we can formulate the product unit waiting time in the queue at operation o as:

$$EW_{po} \approx \sum_{m} EWQ_m \varsigma_{pom} + SUe_{po} + Q_p PRe_{po} \tag{18}$$

with

$$EWQ_m \approx \phi\left(\rho_m, \tilde{c}^2_{IA_m}, \tilde{c}^2_m, s_m\right)\left(\frac{\tilde{c}^2_{IA_m} + \tilde{c}^2_m}{2}\right)\left(\frac{\rho_m^{\sqrt{2(s_m+1)}-1}}{s_m(1-\rho_m)}\right)\tilde{PR}_m \tag{19}$$

Since multiple products p visit machine m through one or more operations o, an expected aggregate lead time at machine m can be determined as a weighted average

$$EW_m \approx EWQ_m + \sum_{p} \frac{\sum_{o} \lambda_p \varsigma_{pom}}{\sum_{p}\sum_{o} \lambda_p \varsigma_{pom}} \left(\sum_{o} \frac{\lambda_p \varsigma_{pom}}{\sum_{o} \lambda_p \varsigma_{pom}} \left(SUe_{po} + Q_p PRe_{po}\right)\right) \tag{20}$$

Note that $\sum_{o} \lambda_p \varsigma_{pom} / \sum_{p}\sum_{o} \lambda_p \varsigma_{pom}$ is the relative importance of product p at machine m, which is independent from the lot size Q_p.

3.4 Objective Functions

We present two objective functions, an operational one and a financial one, that will be subject to a minimization routine. The first objective is an expected aggregate lead time for the overall production system

$$EW \approx \sum_{m} EW_m + \sum_{p} \frac{\lambda_p}{\sum_{p} \lambda_p} WTBT_p \tag{21}$$

The goal is to find a set of lot size values Q_p for each product p that leads to an optimal performance of the entire system from a lead time perspective. However, when costs are involved for holding inventory and setting up machines, the optimal lot size decisions may be different because of a shifted trade-off function. We therefore propose a second objective that incorporates the lead time information from Equations (2) and (20) into a cost function that consists of a setup cost and a lead time related cost. All other cost elements like depreciation, raw materials, disposal, repair, etc. are considered to be irrelevant to the lot size decision. To this end, we list some additional cost parameters: the average cost hc_{po} to hold one unit in stock of product type p at the inbound of operation o during the predefined time bucket of length CT, a fixed setup cost suc_{po} each time the machine related to an operation o of product p is changed over, a number of operators l_{po} required for this setup process and a setup cost suc_m for every sut_m time units that are consumed for setting up machine m. The second, alternative objective function in terms of the expected costs becomes

$$\begin{aligned} EC \approx\ & \sum_{p} \lambda_p \left(hc_{p1} WTBT_p + \sum_{o} \left(hc_{po} \sum_{m} EW_m \varsigma_{pom} + \frac{CT}{Q_p} suc_{po}\right)\right) + \\ & \sum_{m} INT\left[\frac{\sum_{p}\sum_{o} CT(\lambda_p/Q_p) SU_{po} l_{po}}{sut_m} + 1\right] suc_m \end{aligned} \tag{22}$$

The first part in Equation (22) consists of two cost types: expected inventory and setup costs that are associated with an operation o. The inventory costs are obtained by applying expected arrival rates to expected delays like expected WTBT at the first operation and expected queue times at resources m. This is Little's Law. The fixed operation related setup cost suc_{po} is multiplied by the total number of setups in the planning period CT. The second part in the equation takes into account total labor time and cost associated with performing setups. Since not all operators may be involved during the entire setup process that takes SU_{po} time units, l_{po} is allowed to be a non-integer value. It is assumed that a fixed cost suc_m is charged for each integer multiple of sut_m time units required by all the operators for setting up machine m. This is the case when for instance workers are paid full-time.

4 Model Optimization

The goal of this paper is to determine for each product p an optimal value for its lot size Q_p that minimizes either the expected aggregate lead time as defined in Equation (21) or the expected costs as defined in Equation (22). The ARP model as described in Section 3 can be classifed as an integer nonlinear programming problem (INLP). Some characteristics make it difficult to solve:

- Decision variables Q_p are discrete and must be selected within a user defined lower and upper bound (i.e. Q_p^{min} and Q_p^{max}), which constitute physical constraints (for instance due to the capacity limits of the number of available transportation carriers).
- Expected waiting times in the objective functions (Equations 21 and 22) are non-linearly dependent on the utilization level ρ_m.
- Since ρ_m depends on $\tilde{\lambda}_p$, which is equal to λ_p/Q_p, the M constraints in Equation (8) are non-linearly dependent on Q_p.
- There are multiple conditional relationships in $\phi(\rho_m, \tilde{c}^2_{IA_m}, \tilde{c}^2_m, s_m)$.
- The introduction of a staircase function in Equation (22) adds complexity to the search space because of the danger of getting trapped in a local optimum.

The computational complexity not only grows exponentially with the number of discrete variables and the number of decisions within each discrete variable, but also with the number of nonlinear relationships in the model. We use two different algorithms to solve these INLP problems:

- A generic steepest descent method (SD), which is a deterministic search procedure because for a given set of algorithm control parameters, the next step of computation is always exactly known and determined.
- A genetic algorithm, in particular the Differential Evolution method (DE), which is a stochastic search procedure because for a given set of algorithm control parameters, the next step of computation is always unknown and undetermined.

In Section 5 we will compare the optimization performance of both methods. For more details on SD, we refer to Vandaele (1996), while DE is outlined next.

4.1 Differential Evolution Routine

DE is an improved version of a genetic algorithm. It manipulates populations based on the principle of survival of the fittest, but in contrast to evolutionary algorithms, which use a predefined distribution function to drive the mutation, DE uses the difference of randomly sampled pairs of object vectors in such a way that the mutation reflects information of the objective function. Instead of using only local information for each object vector, DE mutates all object vectors with the same universal distribution. The idea is to cover the entire search space and a global optimum may be found.

The method is defined as a parallel direct search method which operates on a population P_G of constant size that is associated with each generation G and consists of N vectors or candidate solutions $\vec{Q}_{n,G}$. Each vector $\vec{Q}_{n,G}$ contains all the decision variables. In ARP, these are the P lot sizes Q_p. An individual lot size value for product p in the population is indicated by $Q_{p,n,G}$. This is briefly summarized as

$$\begin{aligned} P_G &= \left\{\vec{Q}_{1,G}, \vec{Q}_{2,G}, ..., \vec{Q}_{n,G}, ..., \vec{Q}_{N,G}\right\} \\ \vec{Q}_{n,G} &= \left[Q_{1,n,G}, Q_{2,n,G}, ..., Q_{p,n,G}, ..., Q_{P,n,G}\right] \end{aligned}$$

where

$n = 1, 2, ..., N \quad p = 1, 2, ..., P \quad N \geq 4 \quad G = 1, ..., G_{max}$

The DE steps are: selection of the strategy, initialization of the control parameters and the population, mutation and recombination to create new children, checking their upper and lower bounds and building a new generation. The last step consists of constraint handling and evaluation of trial and/or parent vectors.

4.1.1 Strategy

While Storn and Price (1997) suggest ten different strategies of DE, we will choose from the *DE/rand/1/bin*, *DE/rand/2/bin* and the *DE/current-to-rand/1* schemes, which are explained in Subsection 4.1.4.

4.1.2 Control Parameters

The user-defined control parameters, which remain constant during the search process, are the crossover constant CR, the population size NP, the mutation scaling factor F, the coefficient of combination K and the maximum number of generations G_{max}. Their meaning and appropriate values is explained below.

4.1.3 Population

To create the initial population $P_{G=0}$, a different value for each decision variable is randomly generated within its bounds according to

$$Q_{p,n,G=0} = Q_p^{min} + \text{rand}_p\left[0,1\right]\left(Q_p^{max} - Q_p^{min} + 1\right) \tag{23}$$

where $\text{rand}_p\left[0,1\right]$ represents a uniformly distributed random variable that ranges from zero to one. Since the outcome of Equation (23) is a continuous value for the discrete variables Q_p, it is followed by truncation (i.e. rounding down) before the objective function is evaluated.

Although the underlying continuous values are used to create subsequent trial vectors, this procedure maintains the diversity of the population and the robustness of the algorithm because only feasible solutions give feedback to the optimization process (Lampinen and Zelinka, 1999). This is in contrast to the SD procedure, which treats all the discrete variables as continuous during the optimization process and only rounds-off when the search process is finished. This may result in both infeasible and suboptimal solutions, as well as large deviations in the optimal objective function value.

Generating infeasible vectors is not problematic at this initial stage because it is the mutation based on the difference of vectors in combination with the constraint handling method in Subsection 4.1.6 that drives the population towards better solutions. Infeasible solutions are able to improve other, both good and feasible solutions.

4.1.4 Mutation and Recombination

Mutation aims to keep a population robust and to search a new area. DE mutates an object vector by adding the weighted difference of randomly sampled pairs of vectors in the current population P_G. The mutated vector that will be used to build the population for the next generation is denoted by $\vec{V}_{n,G+1}$.

Recombination, or crossover, is complementary to mutation and builds trial vectors out of existing object vector parameters in order to reinforce prior successes. The crossover operation creates a trial vector $\vec{U}_{n,G+1}$ by selecting elements from the target vector $\vec{Q}_{n,G}$ and the mutated donor vector $\vec{V}_{n,G+1}$. The crossover constant CR controls the probability that a trial vector parameter will come from the mutated vector $\vec{V}_{n,G+1}$, instead of from the current vector $\vec{Q}_{n,G}$, and therefore ranges from 0 to 1. We will choose from three mutation schemes: *DE/rand/1/bin*, *DE/rand/2/bin* and *DE/current-to-rand/1*.

In the *DE/rand/1/bin* scheme (further referred to as scheme 1), the population of child or trial vectors $P'_{G+1} = \vec{U}_{n,G+1} = U_{p,n,G+1}$ for each parent or target vector $Q_{p,n,G}$ is created as follows

$$U_{p,n,G+1} = \begin{cases} V_{p,n,G+1} = Q_{p,r_3,G} + F\left(Q_{p,r_1,G} - Q_{p,r_2,G}\right) & \text{if } R\left(0\right) \leq CR \vee p = k \\ Q_{p,n,G} & \text{otherwise} \end{cases}$$

where

$n = 1, ..., N > 3, p = 1, ..., P$

$k \in \{1, ..., P\}$, random decision variable index

$r_1, r_2, r_3 \in \{1, 2, ..., N\}$ randomly selected, but $r_1 \neq r_2 \neq r_3 \neq n$

$CR \in [0, 1]$, $F \in (0, 1+]$, and $R\left(0\right) \in [0, 1]$ is a uniformly random number.

The /1/ in this scheme means that there is one paired difference (with weight F) of randomly chosen vectors that drives the mutation. Effective values for F that scales the step size belongs to the interval $(0, 1+]$, i.e. values slightly larger than one can be used if needed, but do not appear to be productive (Lampinen and Zelinka, 1999). The notation /rand/ means that the donors to be mutated are randomly chosen from the population members. Also, the randomly chosen indices r_1, r_2 and r_3 must be mutually different, and different from n, i.e. the current parent object vector. Consequently, N must be greater than 3. New, random, integer values for r_1, r_2 and r_3 are chosen for each individual candidate solution n. The index k ensures that each child vector will differ from its parent in the previous generation by at least one variable. A new random integer value is assigned to k prior to the construction of each child vector. The binomial scheme (*/bin/*) takes parameters from $\vec{V}_{n,G+1}$ each time when $R(0) \leq CR$, otherwise the parameters come from $\vec{Q}_{n,G}$.

The population of children in the *DE/rand/2/bin* scheme (further referred to as scheme 2) is created in a similar way except that two paired differences of randomly chosen vectors drive the mutation

$$U_{p,n,G+1} = \begin{cases} V_{p,n,G+1} = Q_{p,r_5,G} + F\left(Q_{p,r_1,G} + Q_{p,r_2,G} - Q_{p,r_3,G} - Q_{p,r_4,G}\right) \\ \qquad\qquad\qquad \text{if } R(0) \leq CR \vee p = k \\ Q_{p,n,G} \qquad\qquad\quad \text{otherwise} \end{cases}$$

where

$n = 1, ..., N > 5, p = 1, ..., P$
$k \in \{1, ..., P\}$, random decision variable index
$r_1, r_2, r_3, r_4, r_5 \in \{1, 2, ..., N\}$ randomly selected,
but $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq n$
$CR \in [0, 1]$, $F \in (0, 1+]$, and $R(0) \in [0, 1]$ is a uniformly random number.

In the *DE/current-to-rand/1* (further referred to as scheme 3), child vectors are created as

$$\vec{U}_{n,G+1} = \vec{V}_{n,G+1} = \vec{Q}_{n,G} + K\left(\vec{Q}_{r_3,G} - \vec{Q}_{n,G}\right) + F\left(\vec{Q}_{r_1,G} - \vec{Q}_{r_2,G}\right)$$

where

$r_1, r_2, r_3 \in \{1, 2, ..., N\}$, randomly selected, but $r_1 \neq r_2 \neq r_3 \neq n$
$K \in [-0.5, 1.5]$

The coefficient of combination K is like F also a user-specified parameter that controls the step size. CR does not need to be specified because it is implicitly equal to 1. When $K = 1$, this scheme is equivalent to the *DE/rand/1/bin* with $CR = 1$. When $K = 0$, there is only a simple mutation.

4.1.5 Boundary Check

A parameter of a mutated child vector that fails to lie within the boundary limits is randomly re-set between its original parent value and the violated boundary

$$U_{p,n,G+1} = \begin{cases} Q_p^{min} + R(0)\left(Q_{p,n,G} - Q_p^{min}\right) & \text{if } U_{p,n,G+1} < Q_p^{min} \\ Q_{p,n,G} + R(0)\left(Q_p^{max} - Q_{p,n,G}\right) & \text{if } U_{p,n,G+1} > Q_p^{max} \\ U_{p,n,G+1} & \text{otherwise} \end{cases}$$

4.1.6 Next Generation

To select the vectors for the next generation, a one-to-one competition between each child and its parent is required, first on the level of the model constraints and then on the level of their performance according to the objective function.

Constraints limit the feasible solutions to a subset of the total search space. In the ARP model, we have M constraints from Equation (8). These are now reformulated as being greater than zero when violated: $g_m = \rho_m - 1 \leq 0$. Instead of implementing them as 'soft' constraints by means of penalty functions, which may result in suboptimal or even infeasible solutions due to inappropriate values for the penalty parameters, we prefer to use the alternative constraint handling method from Lampinen (2001) to select the vectors for the next generation:

$$\vec{Q}_{n,G+1} = \begin{cases} \vec{U}_{n,G+1} \text{if} \begin{cases} \begin{cases} \forall m : g_m\left(\vec{U}_{n,G+1}\right) \leq 0 \wedge g_m\left(\vec{Q}_{n,G}\right) \leq 0 \\ \wedge \\ f\left(\vec{U}_{n,G+1}\right) \leq f\left(\vec{Q}_{n,G}\right) \end{cases} \\ \vee \\ \begin{cases} \forall m : g_m\left(\vec{U}_{n,G+1}\right) \leq 0 \\ \wedge \\ \exists m : g_m\left(\vec{Q}_{n,G}\right) > 0 \end{cases} \\ \vee \\ \begin{cases} \exists m : g_m\left(\vec{U}_{n,G+1}\right) > 0 \\ \wedge \\ \forall m : Max\left[g_m\left(\vec{U}_{n,G+1}\right), 0\right] \leq Max\left[g_m\left(\vec{Q}_{n,G}\right), 0\right] \end{cases} \end{cases} \\ \vec{Q}_{n,G} \quad \text{otherwise} \end{cases}$$

A trial or child vector $\vec{U}_{n,G+1}$ when compared with its parent $\vec{Q}_{n,G}$ survives in the next generation when

- all constraints for $\vec{U}_{n,G+1}$ and $\vec{Q}_{n,G}$ are satisfied, but $\vec{U}_{n,G+1}$ has a lower cost or lead time objective value
- all constraints for $\vec{U}_{n,G+1}$ are satisfied but not for $\vec{Q}_{n,G}$
- not all constraints for $\vec{U}_{n,G+1}$ are satisfied, but it violates each constraint equal or less than $\vec{Q}_{n,G}$

If none of these requirements are met, the parent will be retained in the next generation. Note that the objective function is not evaluated until all the constraints are feasible, which reduces computation time and results in a fast convergence towards the feasible regions of the search space. In addition, the routine will not always have to go through all the constraints, which further decrease computation time. This is interesting for our computationally expensive objective and constraint functions.

Comparing a child to only one individual (i.e. its parent), and not to an arbitrary better performing member like in other evolutionary algorithms, ensures that there is no drift towards local optima. Trial vectors that are equally good enter the population to avoid stagnation. The final step is to determine the best population member b

$$\vec{Q}_{b,G+1} = Min\left[f\left(\vec{Q}_{n,G+1}\right)\right]$$

5 Model Results

The first algorithm applied to ARP is the steepest descent method (SD) that is used in Lambrecht et al. (1998). The control parameters are a gradient precision to decide when an element of the gradient vector can be considered small enough to be zero, a step size precision to define the multiplicator applied to the gradient vector, an initial value for this step size, and a stopping precision to decide when an objective function improvement can be considered small enough to quit the search process. Apart from finding appropriate values for these parameters, the main problem with SD is that discrete variables are treated as continuous values. Rounding after optimization may not only lead to sub-optimal results, but also to even infeasible solutions. This is particularly true in ARP, where rounding down a lot size Q_p involves the risk of creating an overutilized resource m, especially in case of relatively small lot size values, high utilization levels and/or high setup times. Therefore, we consistently round up the lot size values found by SD.

The second algorithm applied to ARP is the differential evolution algorithm (DE). Apart from choosing a scheme, the control parameters to be set are N, F, K, CR and G_{max}. Since the purpose of this study is the search for a global optimum for the lot sizes in the ARP model, we use an extremely high value for G_{max} as a stopping criteria. Based on the results in Table 2 (see Research Question 2), we decide to go with $G_{max} = 100,000$. If the optimum is not found after $100,000$ iterations, we can conclude that DE is underperforming with the control parameters used. For the second stopping criteria, we allow only a small deviation $\epsilon = 10^{-7}$ between the best and the worst population member as required by $\left|\frac{\text{Best member - Worst member}}{\text{Worst member}}\right| < \epsilon$. This involves that all members of the population must converge to the same solution before the algorithm is ended. It ensures that results are of a high quality. Appropriate values for the other parameters are based on the guidelines of Storn and Price (1997):

- N should be as small as possible to limit the computation time. $N = 20P$ is appropriate for problems with $P \leq 30$. It can go up to $100P$ for functions with many local optima. When P is large or in case of few local optima, very good results can also be obtained with $N \ll 20P$. However, with $N < 2P$, all the object vectors in the population may tend towards a single, sub-optimal solution (i.e. premature convergence), or no evolution takes place in a still diversified population (i.e. stagnation).

- CR should be close to or equal to 1.

- F and K are closely related. Although K is continuous in the interval $[-0.5, 1.5]$, the range can be limited to $[0, 1]$ for practical problems. Mostly, it is even sufficient to set it to one of the three discrete values: 0, 0.5, 1. When $K \geq 0.5$, F should be in the range $[0.6, 0.8]$; smaller values of F may induce premature convergence. $F > 1$ seems not effective.

In case of premature convergence or stagnation, N and/or F should be increased. When the *DE/current-to-rand/1* is used, K should be decreased to avoid premature convergence, or randomly chosen from the interval [0,1] to avoid stagnation.

Given these technical details about the solution methods, we want to answer our research questions for ARP as described in Section 1.

Scenario	1	2	3	4
P	100	50	25	10
$\sum_p O_p$	640	424	216	86
M	133	133	114	86

Table 1: Dimensions of the 4 Scenarios

Research Question 1 *Is the differential evolution algorithm able to handle large real-life problems?*

To this end, we have generated datasets with realistic problem sizes and features that represent production systems from practice. We have distinguished four scenarios that differ in size with respect to the number of products P, machines M and operations O_p (see Table 1). For each scenario, the number of parallel servers s_m is increased for each machine m until its utilization due to processing time only (i.e. Equation (8) without setup time) is maximum 50%. This step is required to create datasets with enough opportunities for performance improvement by changing the batch size.

Conclusion: From the discussion below, we conclude that DE can be applied to lot size decisions in large real-life stochastic production planning systems.

Research Question 2 *Does the differential evolution algorithm outperforms the steepest descent algorithm, in terms of objective value and computation time? In other words, is DE effective and efficient compared to SD?*

In order to obtain lot size solutions from SD, we have performed an extensive study with different SD parameter settings. In Table 2, we only include the best solution for each scenario in Table 1 and both objective functions from Section 3.4. The number of required function evaluations and the number of iterations give an idea of the computational effort of SD.

			DE				SD
		Scenario	N=20P	N=20P	N=10P	N=10P	
Lead Time	Objective	1	4494	-4.19%	4494	-4.19%	4691
		2	4682	-1.36%	4682	-1.36%	4746
		3	3630	-1.85%	3630	-1.85%	3698
		4	2440	-1.19%	2440	-1.19%	2469
	Evaluation	1	115942898	1727.33%	31158675	391.08%	6344941
		2	14709931	4543.27%	4345846	1271.79%	316801
		3	1189853	251.25%	427878	26.31%	338753
		4	58548	141.32%	39033	60.88%	24262
	Iteration	1	94686		49069		46328
		2	20209		11719		3771
		3	3179		2241		5701
		4	410		495		596
Cost	Objective	1	95326	-3.52%	95326	-3.52%	98805
		2	86330	-0.12%	86330	-0.12%	86434
		3	69424	-0.06%	69424	-0.06%	69467
		4	50918	-0.04%	50918	-0.04%	50939
	Evaluation	1	128663128	27263.72%	33973261	7125.34%	470196
		2	13967309	22026.08%	4366505	6817.13%	63126
		3	1243620	6239.18%	521974	2560.69%	19618
		4	62978	300.27%	31924	102.90%	15734
	Iteration	1	86316		43891		3682
		2	16482		10248		642
		3	2770		2311		366
		4	361		362		403

Table 2: Differential Evolution versus Steepest Descent

Function	Lead Time				Cost			
Scenario	1	2	3	4	1	2	3	4
Mean	9.16	0.38	0.36	0.40	9.14	0.48	0.52	0.50
Variance	2383.49	0.24	0.23	0.24	655.08	0.29	0.25	0.25
Mininmum	0	0	0	0	1	0	0	0
Maximum	404	1	1	1	213	2	1	1

Table 3: Characteristics of the absolute lot size deviations between DE and SD

In the next step, we have applied DE to each scenario and to each objective function from Section 3.4, where different parameter settings, based on the suggestions of Storn and Price (1997) with $N = 20P$ and $N = 10P$, have been used. Since for each case, all these runs independently converges to the same solution with respect to objective and lot size values, we are confident that this solution is near the global optimum. These optimal objective values are displayed in Table 2 as well as the computational effort of an intermediate DE parameter setting (Scheme 1, $CR = 0.99$ and $F = 0.6$). The deviations from the SD results in terms of percentage are included for the number of objective function evaluations and the objective value. In general, increasingly more evaluations are required as the number of decision variables P and the population size N are increased, especially when the cost objective is considered. Another major observation is that the objective value according to SD can always be improved, and usually more when P is high and when the lead time objective is considered. As a result, DE and SD present different lot size values. To get an idea about the magnitude and the extent of these differences, some key measures are calculated in Table 3 for the absolute deviations between the optimal lot size values found by DE and SD for each product p: mean, variance, minimum and maximum. The lot size values according to DE clearly deviate from SD when the number of decision variables is high. Given the large number of required function evaluations, the DE parameter setting used here is not appropriate (see further in Research Question 3).

Conclusion: Compared to the solutions from SD, the objective function can always be significantly improved towards the global optimum (DE is effective), but computation time can be longer when the DE control parameters are inappropriate (DE is not always efficient). The improved performance is more pronounced when more products are involved in ARP.

Research Question 3 *What are efficient control parameters for the differential evolution algorithm?*

Since the required computation time for a DE parameter setting as in Table 2 is not acceptable for management decisions at the intermediate time horizon, we want to select more appropriate sets of DE control parameters. To that end, we first dramatically reduce the population size to $N = 2P$, followed by an extensive study with different values for F and K that are applied to the three schemes as described in Section 4. Note that G_{max} and CR remains at the value of $100,000$ and 0.99 respectively. Tables 6 to 11 in B give an overview of the results, expressed as an increase (+) or decrease (-) in terms of percentage relative to the DE results in Table 2 with $N = 10P$. A quality level number is included to distinguish five groups of DE parameter settings with respect to solution and DE performance.

A first type of DE parameter settings (quality level 5) fastly converges to inferior solutions. It typically occurs for low F values, especially in combination with low P values and the objective of lead time minimization. When Scheme 3 is used with $K = 1$, it occurs for any value of F.

A second type of DE parameter settings (quality level 4) is not able to find good solutions after $100,000$ generations. Due to this very slow convergence rate, these settings are no suitable alternatives for the DE setting in Table 2. It typically occurs for high F values in Scheme 2 and high F values in Scheme 3 with $K = 0$, especially in combination with high P values. When Scheme 3 is used with $K = 0.5$, it usually occurs for any value of F.

A third type of DE settings (quality level 3) is able to find good solutions, but the population is still diversified after $100,000$ generations. This is an indication of either stagnation or a slow convergence rate. Therefore, these settings are not usable in practice because either no information about the global solution is given to know when to quit the search process or it takes too long before all the members in the still diversified population are converged to the optimal solution. It typically occurs for intermediate F values in Scheme 2 and for any F value in Scheme 3 with $K = 0$, especially in combination with low P values.

A fourth type of DE settings (quality level 2) is able to find solutions that are close to the global optimum. These settings can be justified when computation time is critical. It typically occurs for intermediate to high F values in Scheme 1, for $F = 0.4$ in Scheme 2 and for intermediate to high F values in Scheme 3 with $K = [0, 1]$.

Some of the previous settings are exactly equal to the solution from Table 2, both from an objective and lot size point of view (quality level 1). The only setting where this is true for all the scenarios and both objective functions is the *DE/current-to-rand/1* scheme with $F = 0.6$ and $K = [0, 1]$. A significant reduction in the required number of evaluations on top of that makes it the most preferable DE control parameter set. These results are summarized in Table 4.

Function	Scenario	Objective	Evaluations	Iterations
Lead Time	1	0.00%	-91.33%	-65.08%
	2	0.00%	-95.60%	-80.59%
	3	0.00%	-92.06%	-64.97%
	4	0.00%	-88.90%	-50.71%
Cost	1	0.00%	-93.39%	-70.54%
	2	0.00%	-95.27%	-77.96%
	3	0.00%	-94.15%	-71.74%
	4	0.00%	-87.27%	-40.61%

Table 4: Results for different-values with Scheme 3=*DE/current-to-rand/1*, $N = 2P$, $G_{max} = 100,000$, $F = 0.6$, $CR = 0.99$ and $K = [0, 1]$ compared to Scheme 1=*DE/rand/1/bin*, $N = 10P$, $F = 0.6$ and $CR = 0.99$

Conclusion: DE is efficient in the search for the global lot size values in ARP when the *DE/current-to-rand/1* scheme is used with $F = 0.6$ and $K \in [0, 1]$.

Research Question 4 *Is there evidence for the generally postulated convex relationship between lot size and lead time in a multi-product, multi-machine production system?*

From the extensive study above, other important observations can be derived. First of all, we see that the optimal solutions found by DE with $N = 20P$ and $N = 10P$ can also be detected with $N = 2P$. Referring to the guidelines of Storn and Price (1997), we can interpret the required value of N as an indicator of model complexity in terms of the number of local optima that exist. Values up to $100P$ are not uncommon for complex functions with many local optima.

Since Tables 6 to 11 in B show that a population with size $N = 2P$ is often large enough to find the best solution, we can conclude that the lot size model has convexity properties.

A second observation is that both objective functions are rather flat because incremental changes in the lot size values only contribute to an incremental deterioration of the objective value. This corresponds to what we have seen in practice. Different DE runs with different parameter settings give an idea about this sensitivity. This finding is of great value for managerial purposes: some robustness of the search process may be sacrificed in order to promote a faster convergence rate.

A final remark from Tables 6 to 11 in B is that deviations from the optimal objective values are smaller for the cost function. This means that it is easier to find good lot size values for this objective than for lead time reduction, which can be explained by the staircase function.

Conclusion: Results and characteristics of the DE search process provide evidence that a convex relationship exists between the lot size and the lead time in ARP. The cost objective is characterized by a faster convergence rate than the lead time objective.

Scenario	Mean	Variance	Mininmum	Maximum
1	1.5	7.1	0	25
2	1.1	0.8	0	3
3	1.8	1.5	0	4
4	1.8	2.2	0	5

Table 5: Characteristics of the absolute deviations between the optimal lot sizes according to the objective of lead time and cost minimization

Research Question 5 *Does the optimal set of lot size values differ for both objective functions (lead time and costs)?*

As expected, the optimal lot sizes are different for both objective functions because the impact of changes in the lot size value are measured in a different way. Under cost minimization, the lot sizes are usually larger for many products p in our case study, while only a few products p have smaller lot sizes. This can be explained by the cost structure that has penalized the setups relatively more than in the lead time model. To get an idea about the magnitude and the extent of these differences, some key measures are calculated in Table 5 for the absolute deviations between these optimal lot sizes: mean, variance, minimum and maximum.

Conclusion: The optimal lot sizes differ under the objectives of lead time and cost minimization.

6 Conclusions

An Advanced Resource Planning (ARP) model that is based on an existing queueing network, is extended towards parallel servers, multi-period resource schedules and variabilities from rework and breakdown. This model allows selecting appropriate lot sizes in a complex, stochastic production environment.

To this end, we have developed two objective functions, one for the overall expected lead time and one for the overall expected costs related to setting up resources. Both functions depend on the lot size values of multiple products that are processed in a sequence of operations on one or more resources. Different optimal lot size values apply in both objectives.

For different problems with practical relevance, we have shown that in search of the optimal lot sizes, the differential evolution algorithm always outperforms the steepest descent method that has been used in previous, related research. This is particularly the case when many products are present.

We have shown that differential evolution (DE) is able to detect solutions near the global optimum because different algorithm control parameter settings in combination with large population sizes always converge to the same set of lot size values in each case of the study.

The speed of its search process can be enhanced without a reduction in the solution quality by using a smaller population size of twice the number of decision variables in combination with appropriate control parameters. We found that the *DE/current-to-rand/1* with $F = 0.6$ and K randomly chosen between 0 and 1 is the best performing control setting if a relatively fast convergence rate towards the global optimum in the ARP model is desired.

A population that converges towards the global optimum with a size of only twice the number of decision variables is only possible when the objective function is relatively smooth withhout many local optima. Since this is the case in the ARP model, evidence is provided that the convex relationship between the lot size and the objective of lead time or cost minimization holds in a complex production environment with multiple products, multiple operations and multiple resources.

These findings are of great value for practitioners as well: large scale production plants can use the ARP model in combination with the DE optimizer to find lot sizes within an acceptable time limit that further improve their lead time and delivery performance.

A Correction Factor ϕ in the GI/G/m-Model

$$\phi\left(\rho_m, \tilde{c}_{IA_m}^2, \tilde{c}_m^2, s_m\right) = \begin{cases} \left(\frac{4\left(\tilde{c}_{IA_m}^2 - \tilde{c}_m^2\right)}{4\tilde{c}_{IA_m}^2 - 3\tilde{c}_m^2}\right)\phi_1\left(s_m, \rho_m\right) + \left(\frac{\tilde{c}_m^2}{4\tilde{c}_{IA_m}^2 - 3\tilde{c}_m^2}\right)\psi\left(c^2, s_m, \rho_m\right) & \text{if } \tilde{c}_{IA_m}^2 \geq \tilde{c}_m^2 \\ \left(\frac{\tilde{c}_m^2 - \tilde{c}_{IA_m}^2}{2\left(\tilde{c}_{IA_m}^2 + \tilde{c}_m^2\right)}\right)\phi_3\left(s_m, \rho_m\right) + \left(\frac{\tilde{c}_m^2 + 3\tilde{c}_{IA_m}^2}{2\left(\tilde{c}_{IA_m}^2 + \tilde{c}_m^2\right)}\right)\psi\left(c^2, s_m, \rho_m\right) & \text{if } \tilde{c}_{IA_m}^2 \leq \tilde{c}_m^2 \end{cases}$$

where

$$\phi_1\left(s_m, \rho_m\right) = 1 + \gamma\left(s_m, \rho_m\right)$$
$$\phi_3\left(s_m, \rho_m\right) = \left(1 - 4\gamma\left(s_m, \rho_m\right)\right) e^{-\frac{2(1-\rho_m)}{3\rho_m}}$$

$$\psi\left(c^2, s_m, \rho_m\right) = \begin{cases} 1 & c^2 \geq 1 \\ \phi_4\left(s_m, \rho_m\right)^{2\left(1-c^2\right)} & 0 \leq c^2 \leq 1 \end{cases}$$

with

$$\gamma\left(s_m, \rho_m\right) = \min\left\{0.24, \frac{(1-\rho_m)(s_m-1)\left[(4+5s_m)^{0.5}-2\right]}{16 s_m \rho_m}\right\}$$
$$c^2 = \frac{\tilde{c}_{IA_m}^2 + \tilde{c}_m^2}{2}$$
$$\phi_4\left(s_m, \rho_m\right) = \min\left\{1, \frac{\phi_1(s_m,\rho_m) + \phi_3(s_m,\rho_m)}{2}\right\}$$

B Differential Evolution Control Parameter Scenarios

			F				
		Scenario	0.2	0.4	0.6	0.8	0.99
Lead Time	Objective	1	2452.97%	0.89%	**0.00%**	0.04%	0.30%
		2	2965.64%	18.65%	0.04%	**0.00%**	0.14%
		3	3516.87%	231.85%	0.45%	**0.00%**	**0.00%**
		4	2825.68%	1559.49%	0.46%	0.17%	0.07%
	Evaluation	1	-99.71%	-98.41%	-94.59%	-81.10%	-76.91%
		2	-99.46%	-97.18%	-94.40%	-79.62%	-81.62%
		3	-97.88%	-92.48%	-91.22%	-75.02%	-73.73%
		4	-96.98%	-86.90%	-84.39%	-83.63%	-79.50%
	Iteration	1	-99.07%	-94.57%	-77.39%	1.09%	25.98%
		2	-97.99%	-89.29%	-75.24%	10.24%	7.96%
		3	-91.92%	-70.50%	-59.97%	29.81%	51.14%
		4	-88.28%	-48.08%	-29.29%	-8.89%	13.94%
	Quality	1	5	2	**1**	2	2
		2	5	5	2	**1**	2
		3	5	5	2	**1**	**1**
		4	5	5	2	2	2
Cost	Objective	1	653.04%	0.71%	**0.00%**	0.02%	0.30%
		2	543.48%	2.25%	**0.00%**	**0.00%**	0.14%
		3	251.76%	52.73%	0.01%	**0.00%**	**0.00%**
		4	136.82%	79.88%	0.14%	0.07%	0.29%
	Evaluation	1	-99.36%	-98.73%	-94.17%	-77.78%	-76.72%
		2	-99.55%	-97.32%	-94.21%	-78.93%	-75.65%
		3	-98.49%	-88.45%	-91.33%	-78.51%	-68.68%
		4	-97.40%	-88.87%	-83.92%	-76.88%	-65.99%
	Iteration	1	-97.50%	-94.90%	-74.17%	16.29%	30.40%
		2	-98.07%	-88.43%	-73.24%	9.83%	33.63%
		3	-93.21%	-47.73%	-58.50%	10.43%	68.93%
		4	-88.67%	-51.10%	-24.59%	15.47%	76.80%
	Quality	1	5	2	**1**	2	2
		2	5	5	**1**	**1**	2
		3	5	5	2	**1**	**1**
		4	5	5	2	2	2

Table 6: Results for different F-values with Scheme 1=*DE/rand/1/bin*, $N = 2P$, $G_{max} = 100,000$ and $CR = 0.99$ compared to Scheme 1=*DE/rand/1/bin*, $N = 10P$, $F = 0.6$ and $CR = 0.99$

		Scenario	F 0.2	0.4	0.6	0.8	0.99
Lead Time	Objective	1	416.30%	**0.00%**	0.04%	2021.27%	2238.35%
		2	2156.20%	0.04%	0.05%	725.05%	1348.19%
		3	2081.76%	1.38%	**0.00%**	0.16%	176.96%
		4	2253.75%	26.75%	**0.00%**	**0.00%**	2.38%
	Evaluation	1	-98.98%	-94.67%	-95.04%	-57.73%	-62.18%
		2	-98.41%	-94.72%	28.81%	56.00%	58.38%
		3	-96.42%	-90.62%	-34.11%	404.09%	519.13%
		4	-95.35%	-85.60%	-82.27%	-59.53%	-39.47%
	Iteration	1	-96.70%	-78.04%	-79.47%	G_{max}	G_{max}
		2	-94.09%	-77.30%	G_{max}	G_{max}	G_{max}
		3	-86.30%	-59.21%	272.16%	G_{max}	G_{max}
		4	-81.82%	-36.77%	0.61%	226.46%	446.67%
	Quality	1	5	**1**	2	4	4
		2	5	2	3	4	4
		3	5	5	**1**	3	4
		4	5	5	**1**	**1**	5
Cost	Objective	1	230.24%	0.01%	**0.00%**	825.73%	926.23%
		2	387.75%	**0.00%**	0.00%	207.13%	298.01%
		3	264.80%	0.03%	**0.00%**	0.00%	11.19%
		4	62.94%	0.04%	**0.00%**	**0.00%**	0.13%
	Evaluation	1	-98.66%	-95.21%	-95.02%	-56.44%	-56.98%
		2	-98.27%	-94.93%	78.98%	88.06%	84.82%
		3	-98.10%	-91.69%	-41.41%	666.34%	508.38%
		4	-94.50%	-77.06%	-69.23%	-6.54%	0.85%
	Iteration	1	-94.78%	-78.17%	-77.92%	G_{max}	G_{max}
		2	-92.63%	-76.88%	G_{max}	G_{max}	G_{max}
		3	-91.43%	-60.80%	204.11%	G_{max}	G_{max}
		4	-75.97%	5.80%	59.39%	413.81%	649.45%
	Quality	1	5	2	**1**	4	4
		2	5	**1**	3	4	4
		3	5	2	**1**	3	4
		4	5	2	**1**	**1**	2

Table 7: Results for different F-values with Scheme 2=*DE/rand/2/bin*, $N = 2P$, $G_{max} = 100,000$ and $CR = 0.99$ compared to Scheme 1=*DE/rand/1/bin*, $N = 10P$, $F = 0.6$ and $CR = 0.99$

			F				
		Scenario	0.2	0.4	0.6	0.8	0.99
Lead Time	Objective	1	3.28%	69.63%	387.99%	1021.77%	1532.16%
		2	0.11%	1.64%	13.02%	138.26%	303.33%
		3	0.00%	0.00%	0.00%	0.29%	3.35%
		4	3743.54%	4036.25%	2580.24%	366.47%	0.15%
	Evaluation	1	-68.24%	-90.24%	-82.28%	-80.93%	-77.54%
		2	80.95%	41.41%	-40.83%	-31.85%	-13.01%
		3	846.45%	775.56%	842.12%	555.98%	189.45%
		4	31.12%	31.63%	35.33%	36.84%	-62.50%
	Iteration	1	G_{max}	G_{max}	G_{max}	G_{max}	G_{max}
		2	G_{max}	G_{max}	G_{max}	G_{max}	G_{max}
		3	G_{max}	G_{max}	G_{max}	G_{max}	G_{max}
		4	416.77%	418.79%	435.96%	473.94%	110.91%
	Quality	1	4	4	4	4	4
		2	3	4	4	4	4
		3	3	3	3	3	4
		4	5	5	5	5	2
Cost	Objective	1	0.75%	20.47%	280.83%	567.48%	694.88%
		2	0.03%	0.26%	2.62%	14.75%	56.89%
		3	0.00%	0.00%	0.00%	0.03%	0.29%
		4	242.15%	145.89%	125.31%	5.53%	0.02%
	Evaluation	1	-53.86%	-84.06%	-71.62%	-71.43%	-71.74%
		2	103.90%	83.44%	-8.11%	-2.90%	9.24%
		3	783.86%	790.23%	828.37%	779.19%	475.60%
		4	60.26%	61.16%	75.24%	72.76%	-68.83%
	Iteration	1	G_{max}	G_{max}	G_{max}	G_{max}	G_{max}
		2	G_{max}	G_{max}	G_{max}	G_{max}	G_{max}
		3	G_{max}	G_{max}	G_{max}	G_{max}	G_{max}
		4	606.35%	610.50%	673.48%	674.31%	67.13%
	Quality	1	3	4	4	4	4
		2	3	3	4	4	4
		3	3	3	3	3	3
		4	5	5	5	5	2

Table 8: Results for different F-values with Scheme 3=*DE/current-to-rand/1*, $N = 2P$, $G_{max} = 100,000$, $CR = 0.99$ and $K = 0$ compared to Scheme 1=*DE/rand/1/bin*, $N = 10P$, $F = 0.6$ and $CR = 0.99$

			F				
		Scenario	0.2	0.4	0.6	0.8	0.99
Lead Time	Objective	1	4361.75%	339.12%	**0.00%**	110.12%	899.51%
		2	4035.31%	578.44%	**0.00%**	**0.00%**	19.32%
		3	3837.81%	634.89%	**0.00%**	**0.00%**	**0.00%**
		4	3175.81%	928.11%	**0.00%**	**0.00%**	**0.00%**
	Evaluation	1	-99.82%	-99.15%	-91.33%	-78.85%	-65.98%
		2	-99.45%	-98.10%	-95.60%	-2.58%	-25.64%
		3	-97.89%	-95.41%	-92.06%	-54.47%	272.98%
		4	-94.06%	-90.95%	-88.90%	-83.55%	-73.80%
	Iteration	1	-99.44%	-97.30%	-65.08%	G_{max}	G_{max}
		2	-97.99%	-92.94%	-80.59%	369.02%	G_{max}
		3	-91.97%	-82.42%	-64.97%	122.80%	1823.74%
		4	-76.77%	-64.04%	-50.71%	-8.48%	77.58%
	Quality	1	5	5	**1**	4	4
		2	5	5	**1**	**1**	4
		3	5	5	**1**	**1**	**1**
		4	5	5	**1**	**1**	**1**
Cost	Objective	1	1316.33%	202.02%	**0.00%**	32.10%	399.96%
		2	664.15%	130.10%	**0.00%**	**0.00%**	1.11%
		3	278.91%	88.67%	**0.00%**	**0.00%**	**0.00%**
		4	147.49%	38.20%	**0.00%**	**0.00%**	**0.00%**
	Evaluation	1	-99.86%	-99.13%	-93.39%	-71.80%	-60.75%
		2	-99.42%	-98.42%	-95.27%	70.29%	-0.80%
		3	-98.45%	-96.21%	-94.15%	-68.93%	101.81%
		4	-93.24%	-88.83%	-87.27%	-79.75%	-51.51%
	Iteration	1	-99.46%	-96.64%	-70.54%	G_{max}	G_{max}
		2	-97.55%	-93.29%	-77.96%	679.03%	G_{max}
		3	-93.03%	-82.91%	-71.74%	56.34%	921.98%
		4	-70.44%	-50.55%	-40.61%	4.70%	153.31%
	Quality	1	5	5	**1**	4	4
		2	5	5	**1**	**1**	4
		3	5	5	**1**	**1**	**1**
		4	5	5	**1**	**1**	**1**

Table 9: Results for different F-values with Scheme 3=*DE/current-to-rand/1*, $N = 2P$, $G_{max} = 100,000$, $CR = 0.99$ and $K = [0, 1]$ compared to Scheme 1=*DE/rand/1/bin*, $N = 10P$, $F = 0.6$ and $CR = 0.99$

			F				
		Scenario	0.2	0.4	0.6	0.8	0.99
Lead Time	Objective	1	4925.00%	4746.33%	652.03%	5.52%	503.00%
		2	5136.72%	4506.36%	1544.34%	2.49%	1.05%
		3	5053.21%	4858.25%	2444.90%	11.35%	0.13%
		4	3448.10%	4171.36%	1734.87%	15.44%	3.48%
	Evaluation	1	-35.81%	-35.81%	-96.15%	-53.73%	-68.97%
		2	130.11%	130.11%	130.08%	125.76%	56.97%
		3	1068.57%	1068.57%	1068.52%	1021.14%	1030.37%
		4	-75.00%	5023.91%	4992.09%	4732.57%	4954.12%
	Iteration	1	G_{max}	G_{max}	-87.53%	G_{max}	G_{max}
		2	G_{max}	G_{max}	G_{max}	G_{max}	G_{max}
		3	G_{max}	G_{max}	G_{max}	G_{max}	G_{max}
		4	-1.62%	G_{max}	G_{max}	G_{max}	G_{max}
	Quality	1	4	4	5	4	4
		2	4	4	4	4	4
		3	4	4	4	4	3
		4	5	4	4	4	4
Cost	Objective	1	1487.81%	1301.00%	543.60%	2.18%	224.49%
		2	774.95%	764.29%	397.17%	7.37%	1.06%
		3	439.06%	412.57%	275.11%	7.55%	0.01%
		4	231.63%	165.48%	94.18%	13.64%	**0.00%**
	Evaluation	1	-41.13%	-41.13%	-41.14%	-49.60%	-60.47%
		2	129.02%	129.02%	129.02%	127.04%	94.51%
		3	857.91%	857.91%	857.90%	856.95%	23.80%
		4	6164.94%	6164.89%	6164.73%	6163.50%	-57.30%
	Iteration	1	G_{max}	G_{max}	G_{max}	G_{max}	G_{max}
		2	G_{max}	G_{max}	G_{max}	G_{max}	G_{max}
		3	G_{max}	G_{max}	G_{max}	G_{max}	513.28%
		4	G_{max}	G_{max}	G_{max}	G_{max}	120.72%
	Quality	1	4	4	4	4	4
		2	4	4	4	4	4
		3	4	4	4	4	2
		4	4	4	4	4	**1**

Table 10: Results for different F-values with Scheme 3=*DE/current-to-rand/1*, $N = 2P$, $G_{max} = 100,000$, $CR = 0.99$ and $K = 0.5$ compared to Scheme 1=*DE/rand/1/bin*, $N = 10P$, $F = 0.6$ and $CR = 0.99$

			F				
		Scenario	0.2	0.4	0.6	0.8	0.99
Lead Time	Objective	1	4030.34%	2029.66%	212.30%	48.88%	317.99%
		2	4232.78%	1640.28%	324.61%	28.88%	195.60%
		3	3799.05%	2010.74%	358.90%	20.64%	43.89%
		4	3113.96%	2942.74%	1021.96%	308.28%	159.44%
	Evaluation	1	-99.79%	-97.69%	-88.05%	-61.83%	-85.08%
		2	-99.51%	-97.58%	-90.57%	-67.00%	-79.46%
		3	-98.89%	-94.15%	-75.85%	-63.50%	-62.30%
		4	-96.58%	-91.71%	-78.62%	-67.41%	-80.65%
	Iteration	1	-99.35%	-92.64%	-54.65%	64.49%	-33.61%
		2	-98.20%	-90.99%	-63.75%	55.70%	-2.47%
		3	-95.81%	-77.64%	-4.82%	76.80%	103.21%
		4	-86.67%	-67.27%	-10.51%	60.00%	-4.04%
	Quality	1	5	5	5	5	5
		2	5	5	5	5	5
		3	5	5	5	5	5
		4	5	5	5	5	5
Cost	Objective	1	1275.62%	762.53%	140.29%	38.07%	199.04%
		2	691.31%	396.41%	165.92%	5.75%	28.35%
		3	349.57%	215.83%	95.84%	2.79%	11.94%
		4	156.26%	137.84%	66.98%	7.51%	0.78%
	Evaluation	1	-99.85%	-97.81%	-93.01%	-72.95%	-86.76%
		2	-99.49%	-97.12%	-93.04%	-71.44%	-83.23%
		3	-98.80%	-94.84%	-85.18%	-74.32%	-76.15%
		4	-93.23%	-89.61%	-71.04%	-71.67%	-80.27%
	Iteration	1	-99.41%	-91.51%	-71.48%	31.51%	-38.03%
		2	-97.85%	-87.72%	-69.98%	39.97%	-16.50%
		3	-94.63%	-76.63%	-32.37%	29.60%	25.66%
		4	-70.17%	-54.42%	36.74%	33.70%	7.18%
	Quality	1	5	5	5	5	5
		2	5	5	5	5	5
		3	5	5	5	5	5
		4	5	5	5	5	2

Table 11: Results for different F-values with Scheme 3=*DE/current-to-rand/1*, $N = 2P$, $G_{max} = 100,000$, $CR = 0.99$ and $K = 1$ compared to Scheme 1=*DE/rand/1/bin*, $N = 10P$, $F = 0.6$ and $CR = 0.99$

References

Albin, S., 1981. Approximating queues with superposition arrival processes. Ph.D. thesis. Columbia University. Department for Industrial Engineering and Operations Research.

Babu, B., Angira, R., 2002. A differential evolution approach for global optimization of MINLP problems. Proceedings of 4th Asia-Pacific Conference on Simulated Evolution And Learning 2, 880–884.

Bitran, G., Tirupati, D., 1988. Multiproduct queueing networks with deterministic routing: Decomposition approach and the notion of interference. Management Science 34, 75–100.

Buzacott, J., Shanthikumar, J., 1985. On approximate queueing models of dynamic job shops. Management Science 31, 870–887.

Buzacott, J., Shanthikumar, J., 1993. Stochastic Models of Manufacturing Systems. Prentice Hall, Englewood Cliffs, NY.

Hopp, W., Spearman, M., 2000. Factory Physics. The McGraw-Hill Companies, New York.

Karmarkar, U., 1987. Lot sizes, lead times and in-process inventories. Management Science 33, 409–423.

Karmarkar, U., Kekre, S., Kekre, S., 1992. Multi-item batching heuristics for minimization of queueing delays. European Journal of Operational Research 58, 99–111.

Karmarkar, U., Kekre, S., S.Kekre, Freeman, S., 1985. Lot-sizing and lead-time perfromance in a manufacturing cell. Interfaces 15, 1–9.

Kuik, R., Tielemans, P., 2004. Expected time in system analysis of a single-machine multi-item processing center. European Journal of Operational Research 156, 287–304.

Lambrecht, M., Ivens, P., Vandaele, N., 1998. ACLIPS: A capacity and lead time integrated procedure for scheduling. Management Science 44, 1548–1561.

Lampinen, J., 2001. Multi-constrained nonlinear optimization by the differential evolution algorithm, in: 6th On-Line World Conference on Soft Computing in Industrial Applications (WSC6).

Lampinen, J., Zelinka, I., 1999. Mechanical Engineering Design Optimization by Differential Evolution. McGraw-Hill, London. chapter 8. pp. 127–146.

Shanthikumar, J., Buzacott, J., 1981. Open queueing network models of dynamic job shops. International Journal of Production Research 19, 255–266.

Shanthikumar, J., Sumita, U., 1988. Approximations for the time spent in a dynamic job shop with applications to due-date assignment. International Journal of Production Research 26, 1329–1352.

Storn, R., Price, K., 1997. Differential evolution - a simple end efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359.

Vandaele, N., 1996. The Impact of Lot Sizing on Queueing Delays: Multi Product, Multi Machine Models. Ph.D. thesis. Katholieke Universiteit Leuven.

Vandaele, N., Vannieuwenhuyse, I., Cupers, S., 2003. Optimal grouping for a nuclear magnetic resonance scanner by means of an open queueing model. European Journal of Operational Research 151, 181–192.

Whitt, W., 1993. Approximations for the GI/G/m queue. Production and Operations Management 2, 114–161.